A NOVEL APPROACH FOR DISPLAYING DATA ON LCD USING FPGA

Shruti Hathwalia[1], Sansar Chand Sankhyan[2]
Department of EECE Department of EECE
ITMU, Gurgaon ITMU,Gurgaon

***Abstract*-Earlier in microcontroller based approach, every LCD display was associated with a static input. This input was static and cannot be changed by user easily as and when needed. Thus restricting the flexibility to user can have in updating the data. In this paper, we propose to design a prototype where we interface LCD display through FPGA board so as to provide flexibility of data which is being displayed directly to LCD.Thus, this design based Device can prove beneficial for future Consumer Electronics Market. In this design, for serial communication, multi UART with configurable baud rate is implemented. The multi UART and LCD driver are implemented with Verilog language and can be integrated into the FPGA to achieve compact, stable and reliable data transmission.**

Index Terms— **FPGA, LCD**

Tools and equipments used –

a) XILINX SPARTAN 300AN FPGA Board
b) XILINX 12.4 SUITE
c) LCD
d) Seven segment display
e) Bread board
f) Connecting leads and wires

I. INTRODUCTION

Text LCD modules are cheap and easy to interface using a microcontroller or FPGA. A liquid-crystal display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly.

LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images which can be displayed or hidden, such as preset words, digits, and 7-segment displays as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.

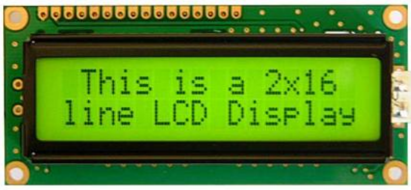

Figure 1.A 2 line x 16 characters LCD module[9]

II. DESIGNING

I. To control an LCD module, you need 11 IO pins to drive an 8-bits data bus and 3 control signals. The 3 control signals are:

- E: enable, or "LCD-select". Active high.
- R/W: read/write. 0 to write, 1 to read.
- RS: register select, 0 for command bytes, 1 for data bytes.

Most of the LCD modules are based on the HD44780 chip or compatible.

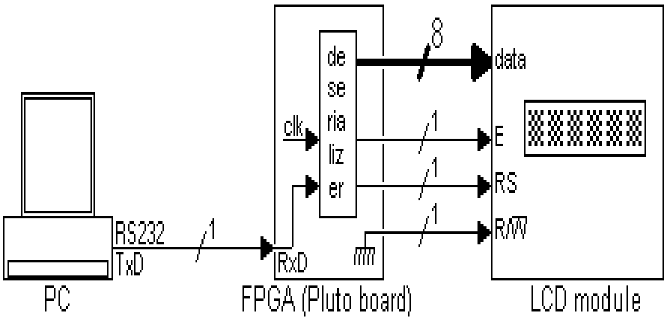

Figure 2.Block diagram of LCD module from an FPGA board [3]

Pluto receives data from the PC serial port, de-serializes it, and sends it to the LCD module. The de-serializer is the same module from the **serial interface** project, so it is just instantiated here. [3]

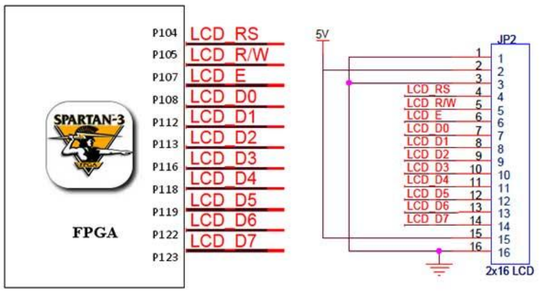

Figure 3.Schematics to Interface 2x16 LCD with Spartan-3 Primer [3]

The principal concept of using hex lies in the data line of LCD display used. Data line of LCD is eight bit long that which implies that 2^8 combinations of decoded values could be passed to LCD enabling incorporation of all the ASCII characters of ASCII table. The system used LCD that has 8 data line and 4 control lines. [2] Control lines namely

I. LCD_RW

II. LCD_EN

III. LCD_RS

IV. LCD_ON

Table 1. ASCII Codes [2]

Alphabet	Hex Value	Alphabet	Hex Value
A	41	P	50
B	42	Q	51
C	43	R	52
D	44	S	53
E	45	T	54
F	46	U	55
G	47	V	56
H	48	W	57
I	49	X	58
J	4A	Y	59
K	4B	Z	5A
L	4C	1	31
M	4D	2	32
N	4E	3	33
O	4F	*	2A

We can display any data of our choice on the LCD just by choosing the desired ASCII code from the above table.

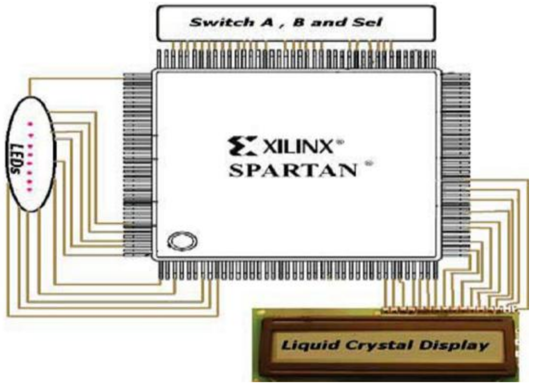

Figure 4.Interfacing of Xilinx SPARTAN board with LCD [1]

III. Verilog code for display-

Here's the verilog code to initialize the LCD module and display 'HELLO WORLD'.

```
module Lcd_Display (clk, lcd_rs, lcd_rw, lcd_e, lcd_4, lcd_5, lcd_6, lcd_7);
          parameter     n = 27;
          parameter     k = 17;
 (* LOC="E12" *)  input          clk;   // synthesis attribute PERIOD clk "100.0 MHz"
          reg    [n-1:0] count=0;
          reg          lcd_busy=1;
          reg          lcd_stb;
          reg      [5:0] lcd_code;
          reg      [6:0] lcd_stuff;
 (* LOC="Y14" *)  output reg    lcd_rs;
 (* LOC="W13" *)  output reg    lcd_rw;
 (* LOC="Y15" *)  output reg    lcd_7;
 (* LOC="AB16" *)  output reg    lcd_6;
 (* LOC="Y16" *)  output reg    lcd_5;
 (* LOC="AA12" *)  output reg    lcd_4;
 (* LOC="AB4" *)  output reg    lcd_e;

 always @ (posedge clk) begin
   count <= count + 1;
   case (count[k+7:k+2])
     0: lcd_code <= 6'b000010;   // function set
     1: lcd_code <= 6'b000010;
     2: lcd_code <= 6'b001100;
     3: lcd_code <= 6'b000000;   // display on/off control
     4: lcd_code <= 6'b001100;
     5: lcd_code <= 6'b000000;   // display clear
     6: lcd_code <= 6'b000001;
     7: lcd_code <= 6'b000000;   // entry mode set
     8: lcd_code <= 6'b000110;
     9: lcd_code <= 6'h24;       // H
    10: lcd_code <= 6'h28;
    11: lcd_code <= 6'h26;       // e
    12: lcd_code <= 6'h25;
    13: lcd_code <= 6'h26;       // l
    14: lcd_code <= 6'h2C;
    15: lcd_code <= 6'h26;       // l
    16: lcd_code <= 6'h2C;
    17: lcd_code <= 6'h26;      // o
    18: lcd_code <= 6'h2F;
    19: lcd_code <= 6'h22;      //
    20: lcd_code <= 6'h20;
    21: lcd_code <= 6'h25;      // W
    22: lcd_code <= 6'h27;
    23: lcd_code <= 6'h26;      // o
    24: lcd_code <= 6'h2F;
    25: lcd_code <= 6'h27;      // r
    26: lcd_code <= 6'h22;
    27: lcd_code <= 6'h26;      // l
    28: lcd_code <= 6'h2C;
    29: lcd_code <= 6'h26;      // d
    30: lcd_code <= 6'h24;
    31: lcd_code <= 6'h22;      // !
    32: lcd_code <= 6'h21;
    default: lcd_code <= 6'b010000;
  endcase
  if (lcd_rw)
    lcd_busy <= 0;
  lcd_stb <= ^count[k+1:k+0] & ~lcd_rw & lcd_busy;  //
clkrate / 2^(k+2)
  lcd_stuff <= {lcd_stb,lcd_code};
  {lcd_e,lcd_rs,lcd_rw,lcd_7,lcd_6,lcd_5,lcd_4}          <=
lcd_stuff;
 end
endmodule
```

IV. Synthesis result-

4.1 UCF Generated

```
# PlanAhead Generated physical constraints

NET "clk" LOC = E12;
NET "lcd_4" LOC = AA12;
NET "lcd_5" LOC = Y16;
NET "lcd_6" LOC = AB16;
NET "lcd_7" LOC = Y15;
NET "lcd_e" LOC = AB4;
NET "lcd_rs" LOC = Y14;
NET "lcd_rw" LOC = W13;
```

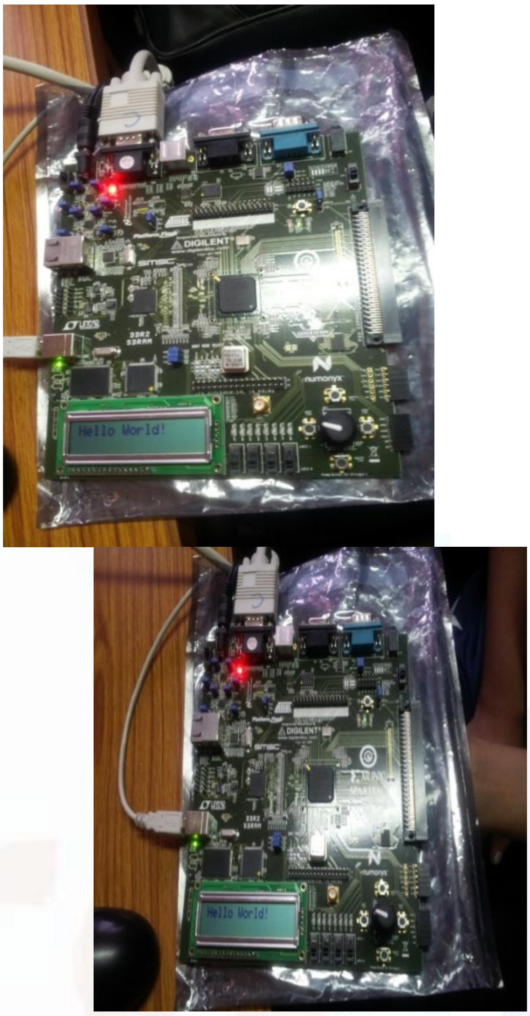

Figure 5. LCD Display “HELLO WORLD” Interfacing on FPGA

V. Conclusion

II. In this paper behavioural level hard -ware descriptive language Verilog is used. Verification of the designed RTL code using simulation techniques, synthesis of RTL code to obtain gate level netlist using Xilinx ISE tool was successfully designed and implemented using Verilog and Xilinx Spatan-300AN Field Programmable Gate Array. Finally we were able to display “HELLO WORLD” on the LCD of FPGA board using interfacing.

References-

[1] Prof. S. Kaliamurthy & Ms. U. Sowmmiya, “VHDL design of arithmetic processor”, Global Journals Inc.(USA) , November 2011.

[2] http://www.ascii.cl/

[3] http://www.fpga4fun.com/TextLCDmodule.html

[4] NAVABI, Z. (1992). *VHDL: Analysis and modeling of digital systems.* McGraw-Hill.

[5] B.Stephen Brown, V.Zvonko, “Fundamentals of digital logic with VHDL Design”2ndEdition,Mc Graw Hill International Edition, 2005.

[6] Digilent, Inc.., Spartan 3E Starter Board, Date Accessed June 2000 http://www.digilentinc.com.
Xilinx Technologies, Xilinx Data Sheet for XC3S100E.h

[7] Fang Yi-yuan; Chen Xue-jun; , "Design and Simulation of UART Serial Communication Module Based on VHDL," Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on , vol., no., pp.1-4, 28-29 May 2011.

[8] Himanshu Patel; Sanjay Trivedi; R. Neelkanthan; V. R. Gujraty; , "A Robust UART Architecture Based on Recursive Running Sum Filter for Better Noise Performance," VLSI Design, 2007. Held jointly with 6th International Conference on Embedded Systems., 20th International Conference on , vol., no., pp.819-823, Jan. 2007.

[9] Hitachi HD44780U (LCD-II) Dot Matrix Liquid Crystal Display Controller/Driver Datasheet, Revision 0.0. Hitachi Ltd.

[10] Neha R. Laddha, A.P.Thakare, International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-1, Issue-6, April 2013.